REMARKS/ARGUMENTS

Claims 1-4, 6-10, and 26 are pending in the present application. Applicants have added claim 26. In this amendment, Applicants have amended claims 1-2 and 9 and canceled claims 5 and 11-25 from further consideration in this application. Applicants are not conceding that the subject matter encompassed by claims 1-2, 5, 9 and 11-25, prior to this Amendment, is not patentable over the art cited by the Examiner. Claims 1-2 and 9 were amended and claims 5 and 11-25 were canceled in this Amendment solely to facilitate expeditious prosecution of the application. Applicants respectfully reserve the right to pursue claims, including the subject matter encompassed by claims 1-2, 5, 9, and 11-25, as presented prior to this Amendment and additional claims in one or more continuing applications. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

The examiner has rejected claims 1-24 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

The Examiner states:

> Claims 1 and 21 recites a method in a data processing system for processing instructions by a processing unit that has or use a standard instruction set and processing only those instructions that use the new instruction set. The claimed suggest using a software program for processing instructions that use the new instruction set. Therefore, claims 1 and 21 are directed to a program per se.
>
> Claim 11 recites a computer program product, which is stored in a computer recordable medium. Although, claim 11 recites a computer recordable medium, this medium can also be a-computer readable media that is directed to non-functional descriptive material used in a data processing system. Specification discloses on pg.28, the computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system. Thus, examiner gives the broadest interpretation for the claimed medium is the form of coded formats. Therefore, claim 11 is directed to a program per se.

Final Office Action dated December 31, 2007, page 3.

Applicants have canceled claims 11-25. Therefore, this rejection has been overcome with respect to claims 11 and 21, and the claims that depend from claims 11 and 25.

Applicants have amended claim 1 to recite: "A method in a computer system for modifying instructions using a processing unit." Claim 1 does not suggest using a software program for processing instructions. Therefore, this rejection has been overcome with respect to claim 1, and the claims that depend from claim 1.

II. 35 U.S.C. § 103, Obviousness

The examiner has rejected claims 1-25 under 35 U.S.C. § 103 as being unpatentable over Zaidi et al., filed December 28, 1999 (US 6,542,981), hereinafter referred to as "*Zaidi*" and further in view of Pechanek et al., filed April 28, 2003 (US Patent No. 6,848,041), hereinafter referred to as "*Pechanek*." This rejection is respectfully traversed.

Zaidi teaches executing a special Reduced Instruction Set Computer (RISC) instruction on a processor to cause an authenticated set of instructions, e.g. microcode, to be transferred to the processor. In this manner, the processor's microcode can be updated. Thus, the new microcode is executed in place of the microcode resident on the processor.

The RISC instruction is a RISC branch instruction designated "br.ia" that requests a special function. If a current instruction is a br.ia instruction, a determination is made regarding whether the function requested by the br.ia instruction is valid. If it is a valid function, the br.ia instruction is executed invoking microcode that is resident in the processor to provide the special function. The invoked microcode causes transfer logic to transfer a new set of microcode instructions to the processor. If the new set of microcode instructions is authenticated, the new microcode instructions are decrypted and stored in the processor.

The Examiner relies on *Zaidi* to teach the features of Applicants' claims except the feature of dynamically remapping the standard instruction set. The Examiner relies on *Pechanek* to teach this feature. Applicants' respectfully disagree that the combination of *Zaidi* and *Pechanek* teaches the features of Applicants' claims

Applicants' claim 1 recites, in part:

> programming a decode unit to decode code using a selected one of the plurality of different instruction maps that was selected in response to a particular reboot of the computer system, wherein a particular new instruction set was created using the selected one of the plurality of different instruction maps;
>
> determining whether or not the particular code is trusted code, wherein code is trusted only when the code resides in an area of the computer system that is trusted to be free of malicious code, and further wherein the area includes a program loader and the plurality of different instruction maps;
>
> in response to determining that the particular code is trusted code: using, by a remapping process in the program loader, the selected one of the plurality of different instruction maps to remap each opcode in the trusted code to new opcode using the particular new instruction set to produce encoded code, wherein the decode unit can decode the encoded code;

in response to determining that the particular code is not trusted code, leaving each opcode in the particular code unchanged, wherein the decode unit cannot decode the particular code;

The combination of *Zaidi* and *Pechanek* does not render Applicants' claims obvious because the combination does not teach or suggest determining whether or not the particular code is trusted code, wherein code is trusted only when the code resides in an area of the computer system that is trusted to be free of malicious code, and further wherein the area includes a program loader and the plurality of different instruction maps.

Applicants claim the particular code is trusted only when the code resides in an area of the computer system that is trusted to be free of malicious code. *Zaidi* teaches authenticating the set of microcode instructions using a one-way hash function, but does not teach an area of the computer system that is trusted to be free of malicious code.

Zaidi also does not teach code being trusted only when it resides in such an area. Further, *Zaidi* does not teach an area that includes a program loader and the plurality of different instructions maps. *Pechanek* does not cure the deficiencies of *Zaidi*. Therefore, the combination of *Zaidi* and *Pechanek* does not render Applicants' claims obvious.

Applicants also claim in response to determining that the particular code is trusted code: using, by a remapping process in the program loader, the selected one of the plurality of different instruction maps to remap each opcode in the trusted code to new opcode using the particular new instruction set to produce encoded code, wherein the decode unit can decode the encoded code. The decode unit has been programmed to decode code using a selected one of the plurality of different instruction maps.

Zaidi teaches using new microcode instructions instead of the resident microcode instructions. *Zaidi* does not teach remapping trusted code using the new microcode instructions to produce encoded code. *Pechanek* does not cure the deficiencies of *Zaidi*. Therefore, the combination of *Zaidi* and *Pechanek* does not render Applicants' claims obvious.

Applicants also claim in response to determining that the particular code is not trusted code, leaving each opcode in the particular code unchanged, wherein the decode unit cannot decode the particular code. The decode unit has been programmed to decode code using a selected instruction map. *Zaidi* teaches updating microcode, but does not teach a decode unit that has been programmed using a selected instruction map that cannot decode particular code that is not trusted. *Pechanek* does not cure the deficiencies of *Zaidi*. Therefore, the combination of *Zaidi* and *Pechanek* does not render Applicants' claims obvious.

Applicants' claim 26 recites:

using, by an encryption algorithm each time the computer system is rebooted, a different one of a plurality of different instruction maps to dynamically remap the standard instruction set to create a new instruction set;

determining whether or not particular code is trusted code, wherein code is trusted only when the code resides in an area of the computer system that is trusted to be free of malicious code, and further wherein the area includes a program loader and the plurality of different instruction maps;

in response to determining that the particular code is trusted code: using, by a remapping process in the program loader, the selected one of the plurality of different instruction maps to remap each opcode in the trusted code to new opcode using the particular new instruction set to produce encoded code; and

in response to determining that the particular code is not trusted code, leaving each opcode in the particular code unchanged.

The combination of *Zaidi* and *Pechanek* does not render claim 26 obvious because the combination does not teach or suggest an area of the computer system that is trusted to be free of malicious code, and further wherein the area includes a program loader and the plurality of different instruction maps. *Zaidi* teaches authenticating and decrypting the new set of microcode, but does not teach an area of the computer system, such as claimed by Applicants. *Pechanek* does not cure the deficiencies of *Zaidi*.

Applicants describe code being trusted only when it resides in the area of the computer system. As discussed above, the combination of *Zaidi* and *Pechanek* does not teach an area of the computer system that is trusted to be free of malicious code. Therefore, the combination does not teach or suggest trusted code as claimed by Applicants. Therefore, the combination of *Zaidi* and *Pechanek* does not render claim 26 obvious.

Zaidi teaches merely the updating of microcode. *Zaidi* does not teach using the selected instruction map to remap each opcode in the trusted code to new opcode in response to determining that the particular code is trusted code. Similarly, *Zaidi* also does not teach leaving each opcode in the particular code unchanged in response to determining that the particular code is not trusted code. Therefore, the combination of *Zaidi* and *Pechanek* does not render claim 26 obvious.

III. Conclusion

It is respectfully urged that the subject application is patentable over *Zaidi* in view of *Pechanek* and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: March 31, 2008

Respectfully submitted,

/Lisa L.B. Yociss/

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant